A diagram of a computer

Description automatically generated

**Backend Development**

1. **Programming Language**:

**Python** for backend development due to its extensive libraries for AI and data handling.

Alternatively, **JavaScript (Node.js)** can be used for real-time data processing.

2. **Web Frameworks**:

**Django** or **Flask** (Python) for a robust backend with AI integration.

**Express.js** (Node.js) for a lightweight and flexible backend.

3. **Database**:

**PostgreSQL** or **MySQL** for structured data like equipment details and research summaries.

**MongoDB** for unstructured or semi-structured data, such as AI-generated suggestions.

**Frontend Development**

1. **Frameworks and Libraries**:

**React.js** or **Vue.js** for building an interactive user interface.

**Three.js** or similar libraries to implement the 360-degree virtual tour functionality.

2. **UI Design Tools**:

Tools like **Figma** or **Adobe XD** for designing user-friendly and visually appealing interfaces.

**AI Tools**

1. **Recommendation System**:

**TensorFlow** or **PyTorch** to build AI models that recommend equipment and research materials tailored to users’ goals.

**Collaborative Filtering** or **Content-Based Filtering** for personalized suggestions.

2. **Natural Language Processing (NLP)**:

**OpenAI API** or **Hugging Face Transformers** for summarizing research papers or responding to user queries about equipment.

3. **Data Collection and Analysis**:

Use **pandas** and **NumPy** for data manipulation.

Employ **scikit-learn** for creating models to analyze usage patterns.

4. **Computer Vision**:

**OpenCV** or **MediaPipe** for interactive 360-degree virtual tours of equipment.

**Cloud Infrastructure**

1. **Hosting Services**:

**AWS**, **Google Cloud Platform (GCP)**, or **Microsoft Azure** for hosting the portal and managing AI models.

2. **Data Storage**:

**Amazon S3** or **Google Cloud Storage** for storing large datasets, research papers, and 360-degree media files.

3. **AI Model Deployment**:

**AWS SageMaker**, **GCP AI Platform**, or **Docker Containers** for deploying AI models in a production environment.

**Authentication and User Management**

1. **Authentication Frameworks**:

**OAuth 2.0** or **JWT (JSON Web Token)** for secure user authentication.

Implement services like **Firebase Authentication** for easier user management.

**APIs for Integration**

1. **Custom APIs**:

Build RESTful APIs using **Django REST Framework** or **FastAPI**.

These APIs can provide endpoints for fetching equipment details, research summaries, and recommendations.

2. **Third-Party APIs**:

Integrate with **research databases** or equipment catalogs via their APIs (if available) to enhance the data pool.

**Implementation Steps**

1. **Define User Journeys**:

Map out key features like login, equipment browsing, and virtual tours.

2. **Database Design**:

Structure your database to store details about equipment, completed research, and user preferences.

3. **Build AI Models**:

Start with basic recommendation algorithms and expand to more advanced ML models.

4. **Develop Frontend and Backend**:

Build and test the user interface and backend APIs.

5. **Integrate AI Models**:

Use APIs to connect your AI models with the portal’s backend.

6. **Test and Deploy**:

Thoroughly test for performance and user experience before deploying the portal.